Available online at www.sciencedirect.com

ScienceDirect

Discrete Optimization 5 (2008) 53–65

DISCRETE OPTIMIZATION

www.elsevier.com/locate/disopt

Nonlinear bipartite matching

Yael Berstein, Shmuel Onn*

Technion — Israel Institute of Technology, 32000 Haifa, Israel

Received 20 March 2006; received in revised form 13 October 2007; accepted 2 November 2007

Abstract

We study the problem of optimizing nonlinear objective functions over bipartite matchings. While the problem is generally intractable, we provide several efficient algorithms for it, including a deterministic algorithm for maximizing convex objectives, approximative algorithms for norm minimization and maximization, and a randomized algorithm for optimizing arbitrary objectives.

MSC: 05A; 15A; 51M; 52A; 52B; 52C; 62H; 68Q; 68R; 68U; 68W; 90B; 90C

Keywords: Matching; Exact matching; Bipartite graph; Birkhoff polytope; Permutation; Permutation matrix; Stochastic matrix; Assignment problem; Combinatorial optimization; Integer programming

1. Introduction

Let $N := \{(i, j) : 1 \le i, j \le n\}$ be the set of edges of the complete bipartite graph $K_{n,n}$. In this article we consider the following broad generalization of the standard linear bipartite matching problem.

Nonlinear bipartite matching. Given positive integers d, n, integer weight functions $w^1, \dots, w^d$ on N, and an arbitrary function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, we find a perfect matching $M \subset N$ minimizing (or maximizing) of the objective function $f(w^1(M), \dots, w^d(M))$ where $w^k(M) := \sum\{w^k(i, j) : (i, j) \in M\}$.

Identifying perfect matchings in $K_{n,n}$ with permutation matrices and weight functions with integer matrices in the usual way, the problem has the following nonlinear integer programming formulation:

$$\text{min or max} \left\{ f(w^1 x, \dots, w^d x) : \sum_{i=1}^{n} x_{i,j} = 1, \sum_{j=1}^{n} x_{i,j} = 1,\ x \in \mathbb{N}^{n\times n} \right\},$$

where $w^k x := \sum_{i=1}^n \sum_{j=1}^n w^k_{i,j} x_{i,j}$ for $k = 1, \dots, d$, and where $\mathbb{N}$ stands for the nonnegative integers.

* Corresponding address: Technion — Israel Institute of Technology, Davidson Faculty of IE & M, Technion City, 32000 Haifa, Israel. Tel.: +972 4 829 4416; fax: +972 4 829 5688.

E-mail addresses: yaelber@tx.technion.ac.il (Y. Berstein), onn@ie.technion.ac.il (S. Onn).

URL: http://ie.technion.ac.il/~onn (S. Onn).

1572-5286/$ - see front matter
doi:10.1016/j.disopt.2007.11.002

The problem can be interpreted as *multiobjective* bipartite matching: given d different linear objective functions $w^1, \ldots, w^d$, the goal is to minimize (or maximize) their "balancing" given by $f(w^1x, \ldots, w^dx)$. The standard linear bipartite matching problem is the special case of $d = 1$ and f is the identity on $\mathbb{R}$.

Our interest in this problem is motivated by several reasons which we now discuss. First, bipartite matching problems, often termed *assignment problems* in the operations research literature, arise naturally in a variety of applications where optimal assignments are to be determined, including personnel assignment (such as assignment of medical students to hospitals for internships), scheduling, facility location, and supply chain management. The standard approach to solving such a problem assumes a linear objective function. However, often this is just an approximation of the real problem which may be nonlinear, and allowing for nonlinear functions results in a much broader expressive power. Moreover, the problem may often involve several parties having different interests and hence different criteria for the quality of a solution, and an objective function of the form $f(w^1x, \ldots, w^dx)$ can compromise the differences and provide a "social optimum". Here is a concrete example.

Example 1.1. Consider the following scheduling problem: assign each of m jobs to any one of p processors, where the processing time of job j if assigned to processor i is $t_{i,j}$, so as to minimize the make-span (last job completion time). Let $n := pm$ and consider $K_{n,n}$ with vertex bipartition $A \uplus B$, with A labeled as $A := \{(i, j) : i = 1, \ldots, p,\ j = 1, \ldots, m\}$ and $B := \{1, \ldots, m\} \uplus B_0$ with B_0 a set of $(p-1)m$ "dummy" vertices. Define p weight vectors $w^1, \ldots, w^p$ on the edges of $K_{n,n}$, by

$$w^i(a, b) := \begin{cases} 0, & \text{if } b \in B_0 \text{ or } a = (k, j) \text{ and } k \neq i; \\ t_{i,j}, & \text{if } a = (i, j) \text{ and } b = j; \\ T, & \text{if } a = (i, j) \text{ and } b = k \neq j, \end{cases} \qquad a \in A, b \in B, i = 1, \ldots, p,$$

where $T := 1 + m \max t_{i,j}$. Let $f(y) = \|y\|_\infty = \max y_i$ be the l_∞ norm on $\mathbb{R}^d$. We claim that a perfect matching minimizing the nonlinear objective $f(w^1x, \ldots, w^px)$ allows us to read off a minimum make-span scheduling. Call a perfect matching *pre-optimal* if each job $b = j \in B \setminus B_0$ is matched to some $a = (i, j) \in A$ (and not to (i, k) with $k \neq j$). By the choice of T, any optimal matching must be pre-optimal. Moreover, any pre-optimal matching defines a scheduling by assigning each job $j \in B \setminus B_0$ to that processor i with $a = (i, j) \in A$ the vertex matched to j. Also, the i-th weight w^ix of a pre-optimal matching x is precisely the total processing time of processor i under the corresponding scheduling. Thus, the criterion of processor i is to minimize its own processing time w^ix. The "social balancing" is the make-span $f(w^1x, \ldots, w^px) = \max w^ix$, and the optimal scheduling is indeed the one corresponding to a perfect matching minimizing the nonlinear objective $\min f(w^1x, \ldots, w^px) = \min \max w^ix$.

A second reason motivating the study of nonlinear bipartite matching arises when viewing it in the more general context of a general nonlinear combinatorial optimization problem—that of optimizing a nonlinear function $f(w^1x, \ldots, w^dx)$ over an arbitrary set S of $\{0, 1\}$-valued vectors. In [7], unifying and extending earlier results of [3,6,8] and the references therein, it was shown that, if the polytope $\operatorname{conv}(S)$ underlying the problem has few edge-directions, then the maximization problem with d fixed and f convex can be solved in strongly polynomial time. This resulted in polynomial time algorithms for the convex maximization for various problems including vector partitioning, matroids, and transportation problems with fixed numbers of suppliers. However, the methods of [3,6–8] do not apply for bipartite matching, since the underlying Birkhoff polytope which is the convex hull of the permutation matrices has exponentially many edge-directions (see Proposition 2.1, Section 2). Since the linear bipartite matching problem is easy to solve, it is natural and important to understand the complexity of convex maximization and general nonlinear optimization for this case.

Finally, a third motivating reason to study nonlinear bipartite matching lies in its interesting connections with various variants and relatives in the literature, including in [1,2,4,5,9,10,12] and the references therein. These variants will be discussed in detail in Section 2. In particular, it turns out that the so-called *exact matching problem*, whose complexity is intriguing and has been long open [5], is closely related to the nonlinear bipartite matching problem. This connection illuminates the exact matching problem from a different, geometric, viewpoint, and may provide some new insights into that problem.

Nonlinear bipartite matching is generally intractable, since already for fixed $d = 1$ (single weight function), the problem of minimizing a family of very simple convex univariate functions $f_u : \mathbb{R} \longrightarrow \mathbb{R}$ defined by $f_u(y) := (y-u)^2$ with u an integer parameter is NP-hard (Proposition 2.3 part 1, Section 2). Therefore, for the most part, the complexity of our results will depend on the unary size of the weights, that is, on $\max |w^k_{i,j}|$. In particular, our algorithms will have polynomial complexity for *binary* weights, that is, with $w^k_{i,j} \in \{0, 1\}$ for all i, j, k. In this case, letting E_k be the support of w^k for each k, the problem becomes that of finding a perfect matching $M \subset N$ maximizing (or minimizing) $f(|M \cap E_1|, \dots, |M \cap E_d|)$. The problem with binary weights is not easy either: the complexity with f an arbitrary function is unknown for any fixed $d \geq 2$; and for variable d it is again NP-hard for minimizing the convex multivariate extension of f_u above, i.e. the family of functions $f_u : \mathbb{R}^d \longrightarrow \mathbb{R}$ defined by $f_u(y) := \sum_{k=1}^{d}(y_k - u_k)^2$ and parameterized by $u \in \mathbb{Z}^d$ (Proposition 2.3 part 2, Section 2).

Clearly, the complexity of the problem depends also on the presentation of the function f: we will mostly assume that f is presented by a *comparison oracle* that, queried on $y, z \in \mathbb{Z}^d$, asserts whether $f(y) \leq f(z)$. This is a broad presentation that reveals little information on the function, making the problem harder to solve. In particular, if d is variable, then already for binary weights and maximizing a convex f, an exponential number of oracle queries is needed (Proposition 2.3 part 3, Section 2).

In spite of these difficulties, we are able to provide the following efficient algorithms for the problem: in the statements below, *oracle-time* refers to the running time plus the number of oracle queries.

Our first theorem provides an efficient algorithm for maximizing convex functions.

Theorem 1.2. *For any fixed d, there is an algorithm that, given any positive integer n, any integer weights $w^1, \dots, w^d$, and any convex function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by the comparison oracle, solves the maximum nonlinear bipartite matching problem in an oracle-time which is polynomial in n and* $\max |w^k_{i,j}|$.

A second theorem provides an efficient randomized algorithm for any function.

Theorem 1.3. *For any fixed d, there is a randomized algorithm that, given any positive integer n, any integer weights $w^1, \dots, w^d$, and any function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by the comparison oracle, solves the nonlinear bipartite matching problem in an oracle-time which is polynomial in n and* $\max |w^k_{i,j}|$.

We also consider the minimum and maximum nonlinear bipartite matching problems where the function f is the l_p norm $\|\cdot\|_p : \mathbb{R}^d \longrightarrow \mathbb{R}$ given by $\|y\|_p = (\sum_{k=1}^{d} |y_k|^p)^{\frac{1}{p}}$ for $1 \leq p < \infty$ and $\|y\|_\infty = \max_{k=1}^{d} |y_k|$. For l_p norm minimization, we give an algorithm which is polynomial in n and $\max |w^k_{i,j}|$ and determines a d-approximative solution for any p and a more accurate, $\sqrt{d}$-approximative solution, for the case of the Euclidean norm $p = 2$ (Theorem 4.1). For l_p norm maximization we give an algorithm which is polynomial even in the bit size of the weights $w^k_{i,j}$ and even if d is variable, and determines a $d^{\frac{1}{p}}$-approximative solution for any p (Theorem 4.2).

We conclude our introduction by pointing out some research directions that arise. First, we consider extending and improving our algorithmic results and the complexity classification of nonlinear optimization over bipartite matchings and relatives. When can efficient approximation algorithms be devised? What is the complexity of the specified multiobjective matching problem with a fixed number d of $\{0, 1\}$-weights with pairwise disjoint supports? When can our randomized algorithm be efficiently de-randomized? Second, we consider extending our results to other nonlinear combinatorial optimization problems where the underlying polyhedra have exponentially many edge-directions, such as matchings in general graphs.

The article proceeds as follows. In Section 2 we discuss various variants and relatives of the problem, survey what is known in the literature about their complexity, and demonstrate the intractability of the problem under various conditions. In Section 3 we discuss convex maximization and prove Theorem 1.2. In Section 4 we discuss approximative norm minimization and maximization and prove Theorems 4.1 and 4.2. Finally, in Section 5 we discuss randomized optimization and prove Theorem 1.3.

2. Variants and intractability

We now discuss various variants and relatives of nonlinear bipartite matching, survey what is known (and unknown) about their complexity, and demonstrate its intractability under various conditions.

First, we note that nonlinear bipartite matching is a special case of a general *nonlinear combinatorial optimization problem*, namely the following: given positive integers d, n, a family $\mathcal{F}$ of subsets of a ground set $\{1, \dots, n\}$, integer weights $w^1, \dots, w^d$ on $\{1, \dots, n\}$, and an arbitrary function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, find $F \in \mathcal{F}$ maximizing (or minimizing)

$f(w^1(F), \ldots, w^d(F))$. In [7], the maximization problem with f convex and d fixed was studied. It was shown that if the number of edge-directions of the polytopes $P^{\mathcal{F}} := \text{conv}\{\mathbf{1}_F : F \in \mathcal{F}\}$ (where $\mathbf{1}_F \in \{0,1\}^n$ denotes the indicator of F) is polynomial in n for a class of families presented by membership oracles, then the problem over families $\mathcal{F}$ in that class can be solved in strongly polynomial oracle-time. This unified and extended earlier results of [6,8] and yielded polynomial time algorithms for convex maximization for various problems including vector partitioning, matroids, and transportation problems with fixed numbers of suppliers. However, for bipartite matching, which is the combinatorial optimization problem over the family $\mathcal{F} \subset 2^N$ of perfect matchings in $K_{n,n}$, the underlying polytope is the Birkhoff polytope

$$P^{\mathcal{F}} = \Pi^n := \left\{ x \in \mathbb{R}_+^{n\times n} : \sum_i x_{i,j} = 1, \sum_j x_{i,j} = 1 \right\}$$

which, as we next show, has exponentially many edge-directions, causing the methods of [6–8] to fail.

Proposition 2.1. *The Birkhoff polytope Π^n has precisely $\frac{1}{2}\sum_{k=2}^n \binom{n}{k}^2 k!(k-1)! \ge \frac{1}{n}\binom{n!}{2}$ edge-directions.*

Proof. Every edge-direction of Π^n is a nonzero minimal-support matrix $x \in \mathbb{R}^{n\times n}$ with zero row-sums and column-sums, and hence (up to scalar multiplication) is the matrix x_C of some circuit $C \subset N$ of $K_{n,n}$, having values ± 1 alternating along the edges of the circuit and 0 elsewhere (see e.g. [7]). We claim that each such circuit matrix x_C is an edge-direction. To see this, let $C = C^+ \uplus C^-$ be the partition of alternating edges of C and let D be a matching in $K_{n,n}$ which perfectly matches all vertices not in C. Let x^+ and x^- be the permutation matrices which are the indicators of the perfect matchings $M^+ := C^+ \cup D$ and $M^- := C^- \cup D$ of $K_{n,n}$. Define a binary weight matrix w as the indicator of $C \cup D$. Then $wx^+ = wx^- = n$ whereas $wx < n$ for any other permutation matrix x. Thus, wx attains its maximum over Π^n precisely at the two vertices x^+ and x^-, and hence $[x^+, x^-]$ is an edge and the difference $x_C = x^+ - x^-$ is an edge-direction. Now, for each $k \ge 2$, the number of $2k$-circuits of $K_{n,n}$ is known and easily seen to be $\frac{1}{2}\binom{n}{k}^2 k!(k-1)!$ (see [7]), and hence the proposition follows. □

Proposition 2.1 shows that, while the methods of [6–8] do apply for transportation problems with fixed numbers of suppliers, they fail for bipartite matching, which is the simplest possible transportation problem – albeit, with variable numbers of suppliers and consumers – and do not lead to a polynomial time algorithm even for maximizing a convex f with fixed d. This state of affairs, along with the easy solvability of the standard linear bipartite matching problem, makes the nonlinear problem for bipartite matching particulary intriguing, and is part of our motivation in raising and studying it herein.

We proceed to discuss variants and relatives of nonlinear bipartite matching and their complexities.

Specified multiobjective bipartite matching. Given d, n, weight functions $w^1, \ldots, w^d : N \longrightarrow \mathbb{Z}$, and integers $u_1, \ldots, u_d$, decide if there is a perfect matching $M \subset N$ satisfying $w^k(M) = u_k$ for all k.

Chandrasekaran et al. considered the problem with a single objective w (fixed $d = 1$) and have shown that already this special case is NP-complete [1]. This raises the question about its complexity in terms of the unary size $\max |w_{i,j}^k|$ of the weights. Indeed, even the case of *binary* weights $w_{i,j}^k \in \{0,1\}$ is not yet understood: for $d = 1$ it was identified as intriguing and mysterious by Papadimitriou and Yanakakis [9,10], and the solutions obtained consequently (first by Karzanov [4] and recently in [12]) are rather sophisticated; for $d = 2$, the complexity is long open; and for variable d it is NP-complete.

The following proposition summarizes the known intractability facts about the specified multiobjective bipartite matching problem.

Proposition 2.2. *The specified multiobjective bipartite matching problem is NP-complete already under the following restrictions:* (1) *fixed $d = 1$ (single weight function);* (2) *binary weights $w_{i,j}^k \in \{0,1\}$.*

Proof. (1) is by reduction from *subset sum* [1]; (2) is by reduction from 3-*dimensional matching*. □

A further specialization of the case of binary weights $w_{i,j}^k \in \{0,1\}$ arises when the w^k have pairwise disjoint supports. This can be formulated as the following particularly appealing "colorful" problem.

Colorful bipartite matching. Given any bipartite graph G with d-colored edge set $E = \biguplus_{k=1}^d E_k$ and $u_1, \ldots, u_d$, decide if there is a perfect matching $M \subseteq E$ containing u_k edges of color k for $k = 1, \ldots, d$.

This problem is a special case of specified multiobjective bipartite matching with binary weights. To see this, note that we may assume that G has the same number n of vertices on each side, making it a subgraph of $K_{n,n}$ with $E \subseteq N$, and $\sum_{k=1}^{d} u_k = n$, else G has no colorful perfect matching; now, letting $w^k \in \{0,1\}^{n\times n}$ be the indicator of E_k for all k, we have that $M \subset N$ is a perfect matching of $K_{n,n}$ with $w^k(M) = u_k$ for all k if and only if M is a perfect matching of G with $|M \cap E_k| = u_k$ for all k.

For $d = 2$ (two colors), this problem is sometimes referred to in the literature as the *exact matching problem*: for $G = K_{n,n}$ it is polynomial time decidable [4,12]; for an arbitrary bipartite graph G there is a randomized algorithm [5], but its deterministic complexity is a longstanding open problem.

Returning to nonlinear bipartite matching, the next proposition describes its intractability under various conditions. By saying that an optimization problem (rather than a decision problem) is NP-hard, we mean, as usual, that there can be no polynomial time algorithm for solving it unless P = NP.

Proposition 2.3. *The following hold for the nonlinear bipartite matching problem with data d, n, weights $w^1, \ldots, w^d \in \mathbb{Z}^{n\times n}$, and function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented explicitly or by a comparison oracle:*

1. *For fixed $d = 1$ (single weight function) and minimizing the simple convex function $f_u : \mathbb{R} \longrightarrow \mathbb{R}$ defined by $f_u(y) := (y-u)^2$ with u an integer input parameter, the problem is already NP-hard.*
2. *For binary weights $w^k_{i,j} \in \{0,1\}$ and minimizing the convex function $f_u : \mathbb{R}^d \longrightarrow \mathbb{R}$ defined by $f_u(y) := \sum_{k=1}^{d}(y_k - u_k)^2$ with $u = (u_1, \ldots, u_d)$ an integer vector, the problem is already NP-hard.*
3. *For binary weights $w^k_{i,j}$ and maximizing a convex f presented by comparison oracle, exponentially many oracle queries are needed, and hence the problem is not solvable in polynomial oracle-time.*

Proof. 1. Given weight w and integer u, there is a perfect matching M with $w(M) = u$ if and only if the minimum value $f_u(w(M)) = (w(M) - u)^2$ of a perfect matching M under f_u is 0. So even computing the optimal objective function *value* enables one to decide the NP-complete problem (1) of Proposition 2.2.

2. Analogously to the proof of part 1 above: given binary weights $w^1, \ldots, w^d$ and vector $u = (u_1, \ldots, u_d)$, there is a perfect matching M with $w^k(M) = u_k$ for all k if and only if the minimum objective value $f_u(w^1(M), \ldots, w^d(M)) = \sum_{k=1}^{d}(w^k(M) - u_k)^2$ of a perfect matching M under f_u is 0. So even computing the optimal objective value enables one to decide the NP-complete problem (2) of Proposition 2.2.
3. Let $d = n^2$, define binary weights $w^{r,s} \in \mathbb{Z}^{n\times n}$ for $1 \le r, s \le n$, with $w^{r,s}_{i,j} := 1$ if $(i,j) = (r,s)$ and $w^{r,s}_{i,j} = 0$ otherwise, and let $f : \mathbb{R}^d \cong \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}$ be any function. Then for any matrix $x \in \mathbb{R}^{n\times n}$ we have $f(w^{1,1}x, \ldots, w^{n,n}x) = f(x_{1,1}, \ldots, x_{n,n}) = f(x)$. Since the permutation matrices (which correspond to perfect matchings) are convexly independent, any assignment of values to the $n!$ permutation matrices can be extended to a convex function f on $\mathbb{R}^{n\times n}$. Thus, to find the permutation matrix maximizing f, the oracle presenting f must be queried on all $n!$ permutation matrices. $\square$

3. Deterministic convex maximization

In this section, we discuss the maximum nonlinear bipartite matching problem for convex functions $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by comparison oracles. We start with some definitions. Here, we will be working with matrices rather than graphs and matchings, so the weights are now integer matrices $w^1, \ldots, w^d \in \mathbb{Z}^{n\times n}$, and the solutions are permutation matrices, which are well known to be precisely the vertices of the Birkhoff polytope of bistochastic matrices (with $\mathbb{R}_+$ the nonnegative reals),

$$\Pi^n = \left\{ x \in \mathbb{R}_+^{n\times n} : \sum_i x_{i,j} = 1, \sum_j x_{i,j} = 1 \right\}.$$

Given weights $w^1, \ldots, w^d$, define a projection $w : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}^d$ mapping matrices x to vectors $w \cdot x$,

$$w \cdot x := (w^1 x, \ldots, w^d x) = \left(\sum_{i,j} w^1_{i,j} x_{i,j}, \ldots, \sum_{i,j} w^d_{i,j} x_{i,j} \right).$$

Define the *multiobjective polytope* (corresponding to $w^1, \ldots, w^d$) to be the projection of Π^n under w,

$$\Pi^n_w := \{w \cdot x = (w^1 x, \ldots, w^d x) \; : \; x \in \Pi^n\} \subset \mathbb{R}^d.$$

Finally, define the *fiber* of any point $y \in \mathbb{R}^d$ to be the polytope $\Pi^n \cap w^{-1}(y)$ consisting of those matrices in the Birkhoff polytope that are projected by w onto y. Thus, a point y is in Π^n_w if and only if its fiber is nonempty; the following lemma asserts that these equivalent conditions can be decided efficiently.

Lemma 3.1. *There is a polynomial time algorithm that, given $d, n, w^1, \ldots, w^d$, and integer $y \in \mathbb{Z}^d$, either asserts $y \notin \Pi^n_w$ and $\Pi^n \cap w^{-1}(y) = \emptyset$ or asserts $y \in \Pi^n_w$ and returns a vertex x of $\Pi^n \cap w^{-1}(y)$.*

Proof. The fiber of any $y = (y_1, \ldots, y_d)$ is the polytope given by the following inequality description,

$$\begin{aligned} \Pi^n \cap w^{-1}(y) &= \{x \in \Pi^n \; : \; (w^1 x, \ldots, w^d x) = (y_1, \ldots, y_d)\} \\ &= \left\{ x \in \mathbb{R}^{n \times n}_+ : \sum_i x_{i,j} = 1, \sum_j x_{i,j} = 1, \; w^k x = y_k \right\}, \end{aligned}$$

so linear programming allows one to efficiently compute a vertex of the fiber or assert that it is empty. □

Lemma 3.1 shows that for any $y \in \mathbb{Z}^d$ it is possible to check efficiently if y is the projection $y = w \cdot x$ of some bistochastic matrix $x \in \Pi^n$, and to find such an x if one exists. We need also to consider the integer analog of this problem: given $y \in \mathbb{Z}^d$ is y the projection $y = w \cdot x$ of some permutation matrix $x \in \operatorname{vert}(\Pi^n)$, and if it is, can we find one such x efficiently? But this problem is *precisely* the specified multiobjective bipartite matching problem: there is a permutation matrix x with $y = w \cdot x$ if and only if there is a perfect matching M with $w^k(M) = y_k$ for $k = 1, \ldots, d$. Unfortunately, as explained in Section 2, the complexity of this problem is open even for fixed $d = 2$. The difficulty is that the fiber $\Pi^n \cap w^{-1}(y)$ of y is not necessarily an integer polytope and it may have some fractional (bistochastic) matrices and some integer (permutation) matrices x as its vertices.

Fortunately, the fibers of *vertices* of Π^n_w are better behaved. The next lemma shows that, if y is any vertex of Π^n_w, then it is possible to find efficiently a permutation matrix x with $y = w \cdot x$.

Lemma 3.2. *Let y be any vertex of the multiobjective polytope Π^n_w. Then the fiber $\Pi^n \cap w^{-1}(y)$ of y is a nonempty integer polytope all of whose vertices are permutation matrices. Thus, the polynomial time algorithm of Lemma 3.1 applied to $y \in \operatorname{vert}(\Pi^n_w)$ returns a permutation matrix x satisfying $y = w \cdot x$.*

Proof. It is well known and easy to see that if Q is the image of a polytope P under an affine map a, then the preimage $P \cap a^{-1}(F) = \{x \in P \; : \; a(x) \in F\}$ of any face F of Q is a face of P. Thus, if y is a vertex of Π^n_w then its fiber $\Pi^n \cap w^{-1}(y)$, which is the preimage under the map w of the face $\{y\}$ of Π^n_w, is a face of Π_n. Therefore, the vertices of the nonempty fiber of y, one of which will be returned by the algorithm of Lemma 3.1, are precisely the vertices of Π^n which are contained in that fiber. □

The next lemma shows that the multiobjective polytope Π^n_w can be constructed efficiently.

Lemma 3.3. *For any fixed d, there is an algorithm that, given n and $w^1, \ldots, w^d \in \mathbb{Z}^{n \times n}$, computes the vertex set $\operatorname{vert}(\Pi^n_w)$ of the multiobjective polytope Π^n_w in a time which is polynomial in n and $\max |w^k_{i,j}|$.*

Proof. Let $u := \max |w^k_{i,j}|$. Then for any permutation matrix x and its projection $y = w \cdot x$, we have $|y_k| = |w^k x| \leq nu$, and therefore y lies in the grid $\{0, \pm 1, \ldots, \pm nu\}^d$. Since each vertex y of Π^n_w is the projection $y = w \cdot x$ of some vertex x of Π^n, which is a permutation matrix, we have $\operatorname{vert}(\Pi^n_w) \subseteq \{0, \pm 1, \ldots, \pm nu\}^d$. For each of the $(2nu + 1)^d$ grid points $y \in \{0, \pm 1, \ldots, \pm nu\}^d$, apply the algorithm of Lemma 3.1 to check if $y \in \Pi^n_w$, and obtain $Y := \{0, \pm 1, \ldots, \pm nu\}^d \cap \Pi^n_w$.

We then have that $\operatorname{vert}(\Pi^n_w) \subseteq Y \subseteq \Pi^n_w$ and therefore the multiobjective polytope Π^n_w is the convex hull of Y. Since convex hulls can be computed in polynomial time for any fixed dimension d, we can efficiently construct Π^n_w, that is, determine all its vertices (and more generally all its faces). □

We can now prove our first theorem, providing an efficient algorithm for convex maximization.

Theorem 1.2. *For any fixed d, there is an algorithm that, given any positive integer n, any integer weights $w^1, \ldots, w^d$, and any convex function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by the comparison oracle, solves the maximum nonlinear bipartite matching problem in an oracle-time which is polynomial in n and* $\max |w^k_{i,j}|$.

Proof. Since f is convex on $\mathbb{R}^d$ and $f(w^1(\cdot), \ldots, w^d(\cdot))$ is convex on $\mathbb{R}^{n\times n}$, and the maximum of a convex function over a polytope is attained at a vertex of the polytope, we have the following equality,

$$\begin{aligned}\max\{f(w^1x, \ldots, w^dx) : x \in \mathrm{vert}(\Pi^n)\} &= \max\{f(w^1x, \ldots, w^dx) : x \in \Pi^n\}\\ &= \max\{f(y) : y \in \Pi^n_w\} = \max\{f(y) : y \in \mathrm{vert}(\Pi^n_w)\}.\end{aligned}$$

Apply the algorithm of Lemma 3.3 and compute $\mathrm{vert}(\Pi^n_w)$. By repeatedly querying the comparison oracle of f, identify a vertex $y^* \in \mathrm{vert}(\Pi^n_w)$ attaining the maximum value $f(y)$. Now apply the algorithm of Lemma 3.1 to y^* and, as guaranteed by Lemma 3.2, obtain a permutation matrix x^* in the fiber of y^*, so that $y^* = w \cdot x^* = (w^1x^*, \ldots, w^dx^*)$ and $f(w^1x^*, \ldots, w^dx^*) = f(y^*)$. Since y^* attains the maximum on the right-hand side of the equation above, x^* attains the maximum on the left-hand side. Thus, the perfect matching of $K_{n,n}$ corresponding to the permutation matrix x^* is optimal. $\square$

The most time consuming part of the algorithm underlying Theorem 1.2 is the repeated use of linear programming for testing fibers of points in the grid $\{0, \pm 1, \ldots, \pm nu\}^d$ to construct $\mathrm{vert}(\Pi^n_w)$. There are various ways of improving the algorithm in practice, but they do not seem to improve the worst case complexity. We now describe such a variant of the algorithm which will usually be much faster since it will typically test the fibers of some but not all points in the grid.

A variant of the convex maximization algorithm.

1. Find the smallest grid containing $\mathrm{vert}(\Pi^n_w)$ by solving, for $k = 1, \ldots, d$, the two linear programs
$$s_k := \min\left\{w^kx : \sum_i x_{i,j} = \sum_j x_{i,j} = 1,\ x \geq 0\right\}, \quad t_k := \max\left\{w^kx : \sum_i x_{i,j} = \sum_j x_{i,j} = 1,\ x \geq 0\right\};$$
then $\mathrm{vert}(\Pi^n_w)$ is contained in the grid $Z := \{y \in \mathbb{Z}^d : s_k \leq y_k \leq t_k,\ k = 1, \ldots, d\}$.
2. By repeatedly querying the comparison oracle of f, order the grid points by nonincreasing value under f and label them $y^1, \ldots, y^{|Z|}$, so that $Z = \{y^1, \ldots, y^{|Z|}\}$ and $f(y^1) \geq \ldots \geq f(y^{|Z|})$.
3. Apply the algorithm of Lemma 3.1 to test the fibers of each y^i in order, until the first y^k for which the vertex x^* of its fiber $\Pi^n \cap w^{-1}(y^k)$ returned by the algorithm is a permutation matrix.
4. Output the perfect matching of $K_{n,n}$ corresponding to the permutation matrix x^*.

We claim that x^* is an optimal solution to the maximum convex bipartite matching problem. Indeed, note that $f^* := \max\{f(y) : y \in \Pi^n_w\} = \max\{f(y) : y \in \mathrm{vert}(\Pi^n_w)\}$ equals the optimal objective function value $\max\{f(w^1x, \ldots, w^dx) : x \in \mathrm{vert}(\Pi^n)\}$ (see proof of Theorem 1.2); let $y^m \in \mathrm{vert}(\Pi^n_w) \subseteq Z$ be a vertex achieving that maximum value $f(y^m) = f^*$; by Lemma 3.2, the algorithm of Lemma 3.1 applied to $y^m \in \mathrm{vert}(\Pi^n_w)$ returns a permutation matrix and so $k \leq m$; this implies $f^* \geq f(y^k) \geq f(y^m) = f^*$ and hence $f^* = f(y^k) = f(w^1x^*, \ldots, w^dx^*)$; therefore x^* achieves the optimal objective function value.

We end this section with an example of a maximum convex bipartite matching problem, demonstrating all notions and algorithms discussed above, some of which will be also used in later sections.

Example 3.4. Consider the maximum convex bipartite matching problem with the following data:

$$d = 2, \qquad n = 4, \qquad w^1 = \begin{pmatrix}1&0&0&0\\1&0&0&1\\1&1&0&0\\0&0&0&1\end{pmatrix}, \qquad w^2 = \begin{pmatrix}1&1&0&1\\0&0&1&1\\0&0&1&0\\1&1&0&0\end{pmatrix}, \qquad f(y) = y_1^2 + y_2^2.$$

By solving the linear programs minimizing and maximizing w^kx over Π^4 for $k = 1, 2$ (step 1 of the algorithm above) we get $s_1 = s_2 = 0$, $t_1 = 3$, $t_2 = 4$, and so the smallest grid containing $\mathrm{vert}(\Pi^4_w)$ is $Z := \{y \in \mathbb{Z}^2 : 0 \leq y_1 \leq 3, 0 \leq y_2 \leq 4\} \subsetneq \{0, \pm 1, \ldots, \pm 4\}^2$ which contains 20 points. Fig. 1 depicts this grid and indicates the objective

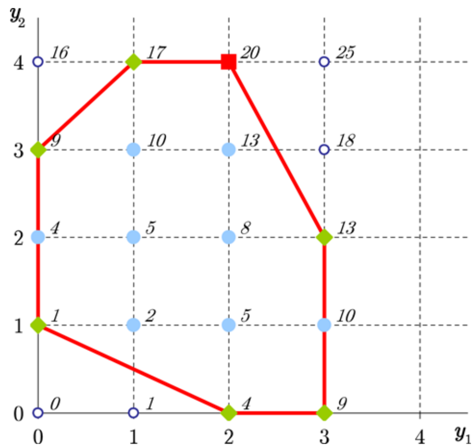

Fig. 1. Example 3.4 and the polytope Π_w^4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

function value $f(y) = y_1^2 + y_2^2$ of each grid point. Ordering the points by decreasing value under f (step 2 above) we get $y^1 = (3, 4)$, $y^2 = (2, 4)$, $\ldots$, $y^{20} = (0, 0)$. Testing fibers of the y^i in order (step 3 above), the fiber of y^1 is found to be empty whereas the fiber of y^2 is nonempty and is the first for which the algorithm of Lemma 3.1 returns a permutation matrix

$$x^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Thus, the corresponding matching $M^* := \{(1, 1), (2, 4), (3, 3), (4, 2)\}$ of $K_{4,4}$ is an optimal solution.

Fig. 1 also shows the multiobjective polytope Π_w^4 and its vertex set $\mathrm{vert}(\Pi_w^4)$ computed by the algorithm of Lemma 3.3: blue circles are non-vertex grid points in Π_w^4 and green diamonds are vertices of Π_w^4. The optimal point $y^2 = (2, 4)$, which is found either by the algorithm above or by the algorithm of Theorem 1.2 is the vertex of Π_w^4 attaining its maximum value under f and is a red square. Of particular interest is the blue point $y = (1, 2)$ whose fiber $\Pi^n \cap w^{-1}(y)$ is a non-integer polytope with 30 vertices (more than the 24 of the Birkhoff polytope upstairs!), all of which are fractional, such as

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0.5 & 0.5 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0.25 & 0.25 & 0.5 \\ 0.25 & 0.75 & 0 & 0 \\ 0.25 & 0 & 0.75 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0.2 & 0.4 & 0.4 \\ 0 & 0.8 & 0 & 0.2 \\ 0.4 & 0 & 0.6 & 0 \\ 0.6 & 0 & 0 & 0.4 \end{pmatrix},$$

indicating the difficulty of the specified multiobjective and colorful bipartite matching problems.

4. Approximative norm optimization

Consider any discrete optimization problem with a finite set S of feasible solutions and nonnegative objective function $g : S \longrightarrow \mathbb{R}_+$ to be minimized or maximized and let $s^* \in S$ be any optimal solution. Then an *r-approximative solution* is any feasible solution $s \in S$ satisfying $\frac{1}{r} g(s^*) \le g(s) \le r g(s^*)$.

In this section, building on the tools and results of Section 3, we provide approximative algorithms for the minimum and maximum nonlinear bipartite matching problems where the function f is the l_p norm $\|\cdot\|_p : \mathbb{R}^d \longrightarrow \mathbb{R}$ given by $\|y\|_p = (\sum_{k=1}^d |y_k|^p)^{\frac{1}{p}}$ for $1 \le p < \infty$ and $\|y\|_\infty = \max_{k=1}^d |y_k|$. To keep our results general and allow the treatment of fractional and even nonrational p, we will still assume that f is presented by a comparison oracle. Of course, for the most common values $p = 1, 2, \infty$ such an oracle is realizable in polynomial time in the rest of the data; moreover, for any integer p, by computing and comparing the integer valued p-th power $\|y\|_p^p$ of the norm instead of the norm itself, such an oracle is realizable in polynomial time in the rest of the data and $\lceil \log p \rceil$.

4.1. Minimization

The following theorem provides an efficient approximative algorithm for minimizing the l_p norm.

Theorem 4.1. *For any fixed d, there is an algorithm that, given any n, any $1 \le p \le \infty$, and any nonnegative integer weights $w^1, \ldots, w^d$, determines a d-approximative solution to the minimum nonlinear bipartite matching problem with $f = \|\cdot\|_p$, in an oracle-time which is polynomial in n and $\max w_{i,j}^k$. For $p = 2$ (the Euclidean norm), the algorithm determines a more accurate, $\sqrt{d}$-approximative, solution.*

Proof. The algorithm is the following: apply the algorithm of Lemma 3.3 and construct the vertex set $\mathrm{vert}(\Pi_w^n)$ of the multiobjective polytope. Using the comparison oracle of f, identify a vertex $\hat{y} \in \mathrm{vert}(\Pi_w^n)$ attaining minimum value $\|y\|_p$. Now apply the algorithm of Lemma 3.1 to $\hat{y}$ and, as guaranteed by Lemma 3.2, obtain a permutation matrix $\hat{x}$ in the fiber of $\hat{y}$, so that $\hat{y} = w \cdot \hat{x}$. Output the perfect matching of $K_{n,n}$ corresponding to the permutation matrix $\hat{x}$.

We now show that this provides the claimed approximation. Let x^* be the permutation matrix corresponding to an optimal perfect matching and let $y^* := w \cdot x^*$ be its projection. Let y' be a point on the boundary of Π_w^n satisfying $y' \le y^*$. By Carathéodory's theorem (on the boundary), y' is a convex combination $y' = \sum_{i=1}^r \lambda_i y^i$ of some $r \le d$ vertices of Π_w^n and some coefficients $\lambda_i \ge 0$ with $\sum_{i=1}^r \lambda_i = 1$. Let t be an index for which $\lambda_t = \max \lambda_i$. Then $\lambda_t \ge \frac{1}{r} \sum_{i=1}^r \lambda_i = \frac{1}{r} \ge \frac{1}{d}$.

Since the weights w^k are nonnegative, we find that so are y' and the y^i and hence we obtain

$$\begin{aligned} f(w^1\hat{x}, \ldots, w^d\hat{x}) &= \|\hat{y}\|_p \le \|y^t\|_p \le d\lambda_t \cdot \|y^t\|_p = d \cdot \|\lambda_t y^t\|_p \\ &\le d \cdot \left\| \sum_{i=1}^r \lambda_i y^i \right\|_p = d \cdot \|y'\|_p \le d \cdot \|y^*\|_p = d \cdot f(w^1 x^*, \ldots, w^d x^*). \end{aligned}$$

This proves that $\hat{x}$ provides a d-approximative solution for any $1 \le p \le \infty$. Now consider the case of the Euclidean norm $p = 2$. By Cauchy–Schwartz, $1 = (\sum_{i=1}^r 1 \cdot \lambda_i)^2 \le \sum_{i=1}^r 1^2 \sum_{i=1}^r \lambda_i^2 = r \sum \lambda_i^2 \le d \sum \lambda_i^2$. Find s with $\|y^s\|_p = \min \|y^i\|_p$ and recall that the y^i's are nonnegative. We then have the inequality

$$\begin{aligned} f^2(w^1\hat{x}, \ldots, w^d\hat{x}) &= \|\hat{y}\|_2^2 \le \|y^s\|_2^2 \le \left(d \sum_{i=1}^r \lambda_i^2 \right) \cdot \|y^s\|_2^2 \le d \sum_{i=1}^r \lambda_i^2 \cdot \|y^i\|_2^2 \\ &\le d \cdot \left\| \sum_{i=1}^r \lambda_i y^i \right\|_2^2 = d \cdot \|y'\|_2^2 \le d \cdot \|y^*\|_2^2 = d \cdot f^2(w^1 x^*, \ldots, w^d x^*) \end{aligned}$$

which proves that in this case, as claimed, $\hat{x}$ provides moreover a $\sqrt{d}$-approximative solution. $\square$

4.2. Maximization

The following theorem provides an approximative algorithm for maximizing the l_p norm that runs in a time which is polynomial even in the bit size of the weights $w_{i,j}^k$ and even if d is variable.

Theorem 4.2. *There is an algorithm that, given any d, any n, any $1 \le p \le \infty$, and any nonnegative integer weights $w^1, \ldots, w^d$, determines a $d^{\frac{1}{p}}$-approximative solution to the maximum nonlinear bipartite matching problem with $f = \|\cdot\|_p$, in oracle-time which is polynomial in d, n, and $\max \lceil \log w_{i,j}^k \rceil$.*

Proof. The algorithm is the following: for $k = 1, \ldots, d$ solve the linear programming problem

$$\max \left\{ w^k x \ : \sum_i x_{i,j} = 1, \sum_j x_{i,j} = 1,\ x \geq 0 \right\},$$

obtain an optimal vertex x^k of Π^n, and let $y^k := w \cdot x^k$ be its projection. Using the comparison oracle of f, find r with $\|y^r\|_p = \max_{k=1}^d \|y^k\|_p$. Output the perfect matching of $K_{n,n}$ corresponding to x^r.

We now show that this provides the claimed approximation. Let s satisfy $\|y^s\|_\infty = \max_{k=1}^d \|y^k\|_\infty$. First, we claim that any $y \in \Pi^n_w$ satisfies $\|y\|_\infty \leq \|y^s\|_\infty$. To see this, choose any point $x \in \Pi^n \cap w^{-1}(y)$ in the fiber of y so that $y := w \cdot x$, let t satisfy $y_t = \|y\|_\infty = \max_{k=1}^d y_k$, and recall that the w^k (and hence the y^k are all nonnegative). Then, as claimed, we get

$$\|y\|_\infty = y_t = w^t x \leq \max\{w^t x \ : \ x \in \Pi^n\} = w^t x^t = y^t_t \leq \|y^t\|_\infty \leq \|y^s\|_\infty.$$

Let x^* be an optimal permutation matrix and let $y^* := w \cdot x^*$ be its projection. Consider first the case $p = \infty$. Then $\|y^r\|_\infty = \max_{k=1}^d \|y^k\|_\infty = \|y^s\|_\infty$ and hence, by the claim just proved, we have

$$f(w^1 x^*, \ldots, w^d x^*) = \|y^*\|_\infty \leq \|y^s\|_\infty = \|y^r\|_\infty = f(w^1 x^r, \ldots, w^d x^r) \leq f(w^1 x^*, \ldots, w^d x^*).$$

Therefore the equality holds all along, and x^r provides an exact optimal solution, or in other words, a 1-approximative solution, agreeing with the statement of the theorem with $d^{\frac{1}{\infty}} = 1$ for $p = \infty$. Next, consider the case of any $1 \leq p < \infty$. Then we have the following inequality which completes the proof,

$$\begin{aligned} f^p(w^1 x^*, \ldots, w^d x^*) &= \|y^*\|_p^p = \sum_{k=1}^d |y^*_k|^p \leq d \cdot \|y^*\|_\infty^p \leq d \cdot \|y^s\|_\infty^p \\ &\leq d \cdot \sum_{k=1}^d |y^s_k|^p = d \cdot \|y^s\|_p^p \leq d \cdot \|y^r\|_p^p = d \cdot f^p(w^1 x^s, \ldots, w^d x^s). \quad \square \end{aligned}$$

5. Randomized nonlinear optimization

In this section, we provide a randomized algorithm for nonlinear bipartite matching for any function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by a comparison oracle. By this we mean an algorithm that has access to a random bit generator, and on any input outputs the optimal solution with a probability of at least half.

By adding to each $w^k_{i,j}$ a suitable positive integer v and replacing the function f by the function that maps each $y \in \mathbb{R}^d$ to $f(y_1 - nv, \ldots, y_d - nv)$ if necessary, we may and will assume without loss of generality throughout this section that the given weights are nonnegative, $w^1, \ldots, w^d \in \mathbb{N}^{n \times n}$.

Recall that $\mathrm{vert}(\Pi^n)$ is the set of $n \times n$ permutation matrices, and let $Y := \{w \cdot x \ : \ x \in \mathrm{vert}(\Pi^n)\}$ be the set of all projections $y = w \cdot x = (w^1 x, \ldots, w^d x) \in \mathbb{N}^d$ of permutation matrices x.

In this section, we will be working with polynomials with integer coefficients in the $n^2 + d$ variables $a_{i,j}$, $i, j = 1, \ldots, n$ and b_k, $k = 1, \ldots, d$. Define an $n \times n$ matrix A whose entries are monomials, by

$$A_{i,j} := a_{i,j} \prod_{k=1}^d b_k^{w^k_{i,j}}, \quad i, j = 1, \ldots, n.$$

Also, for each matrix $x \in \mathbb{N}^{n \times n}$ and vector $y \in \mathbb{N}^d$, define the following monomials,

$$a^x := \prod_{i=1}^n \prod_{j=1}^n a_{i,j}^{x_{i,j}}, \qquad b^y := \prod_{k=1}^d b_k^{y_k}.$$

As usual, the *sign* of a permutation matrix x is defined by the parity of any number of transpositions whose product gives x, with $\mathrm{sign}(x) = 1$ when this number is even and $\mathrm{sign}(x) = -1$ when it is odd. Finally, for each $y \in \mathbb{N}^d$ define the following polynomial in the variables $a = (a_{i,j})$ only, by

$$g_y(a) := \sum \left\{ \mathrm{sign}(x) a^x \ : \ x \in \mathrm{vert}(\Pi^n),\ w \cdot x = y \right\}.$$

We then have the following identity expanding the determinant of A in terms of the $g_y(a)$,

$$\det(A) = \sum_{x \in \mathrm{vert}(\Pi^n)} \mathrm{sign}(x) \prod_{i,j} A_{i,j}^{x_{i,j}} = \sum_{x \in \mathrm{vert}(\Pi^n)} \mathrm{sign}(x) a^x b^{w \cdot x} = \sum_{y \in Y = w \cdot \mathrm{vert}(\Pi^n)} g_y(a) b^y.$$

Next, we consider integer substitutions to the variables $a_{i,j}$. Under such substitutions, each $g_y(a)$ becomes an integer and $\det(A) = \sum_{y \in Y} g_y(a) b^y$ becomes a polynomial in the variables $b = (b_k)$ only. Given such a substitution, let $\hat{Y} := \{y \in Y : g_y(a) \neq 0\}$ be the *support* of $\det(A)$, that is, the set of exponents of monomial b^y appearing with nonzero coefficient in $\det(A)$.

The next proposition concerns substitutions of independent identical random variables uniformly distributed on the set of integers $\{1, 2, \ldots, s\}$, under which $\hat{Y}$ becomes a random subset of Y.

Proposition 5.1. *Suppose that independent identical random variables uniformly distributed on the set* $\{1, 2, \ldots, s\}$ *are substituted for the* $a_{i,j}$, *and let* $\hat{Y} = \{y \in Y : g_y(a) \neq 0\}$ *be the random support of* $\det(A)$. *Then, for every* $y \in Y = \{w \cdot x : x \in \mathrm{vert}(\Pi^n)\}$, *the probability that* $y \notin \hat{Y}$ *is at most* $\frac{n}{s}$.

Proof. Consider any $y \in Y$ and consider $g_y(a)$ as a polynomial in the variables $a = (a_{i,j})$. Since $y = w \cdot x$ for some permutation matrix, there is as least one term $\mathrm{sign}(x)a^x$ in $g_y(a)$. Since distinct permutation matrices x give distinct monomials a^x, no cancellations occur among the terms $\mathrm{sign}(x)a^x$ in $g_y(a)$. Thus, $g_y(a)$ is a nonzero polynomial of degree n. The claim now follows from a lemma of Schwartz [11] stating that the substitution of independent identical random variables uniformly distributed on $\{1, 2, \ldots, s\}$ into a nonzero multivariate polynomial of degree n is zero with probability at most $\frac{n}{s}$. $\square$

The next lemma shows that, given $a_{i,j}$, the support $\hat{Y}$ of $\det(A)$ is polynomial time computable.

Lemma 5.2. *For any fixed* d, *there is an algorithm that, given* n, $w^1, \ldots, w^d \in \mathbb{N}^{n \times n}$, *and substitutions* $a_{i,j} \in \{1, 2, \ldots, s\}$, *computes* $\hat{Y} = \{y \in Y : g_y(a) \neq 0\}$ *in polynomial time in* n, $\max w_{i,j}^k$ *and* $\lceil \log s \rceil$.

Proof. For each y, let $g_y := g_y(a)$ be the fixed integer obtained by substituting the given integers $a_{i,j}$. Put $u = n \cdot \max w_{i,j}^k$ and $Z = \{0, 1, \ldots, u\}^d$. Then $\hat{Y} \subseteq Y \subseteq Z$ and hence $\det(A) = \sum_{y \in Z} g_y b^y$ is a polynomial in d variables $b = (b_k)$ involving at most $|Z| = (u+1)^d$ monomials. For $t = 1, 2, \ldots, (u+1)^d$, consider the substitution $b_k := t^{(u+1)^{k-1}}$, $k = 1, \ldots, d$. Let $A(t)$ be the integer matrix obtained from A by this substitution along with the substitution of the given $a_{i,j}$. Then each entry of $A(t)$ satisfies

$$A(t)_{i,j} = a_{i,j} \prod_{k=1}^{d} (t^{(u+1)^{k-1}})^{w_{i,j}^k} \leq s \prod_{k=1}^{d} (((u+1)^d)^{(u+1)^{k-1}})^{\frac{u}{n}} \leq s(u+1)^{d(u+1)^{d+1}}$$

and hence its bit size $1 + \log A(t)_{i,j} = O(u^{d+1} \log(su))$ is polynomially bounded in n, $\max w_{i,j}^k$, $\lceil \log s \rceil$. Therefore the integer number $\det(A(t))$ can be computed in polynomial time by Gaussian elimination. So we obtain the following system of $(u+1)^d$ equations in $(u+1)^d$ variables g_y, $y \in Z = \{0, 1, \ldots, u\}^d$,

$$\det(A(t)) = \sum_{y \in Z} g_y \prod_{k=1}^{d} b_k^{y_k} = \sum_{y \in Z} t^{\sum_{k=1}^{d} y_k (u+1)^{k-1}} \cdot g_y, \quad t = 1, 2, \ldots, (u+1)^d.$$

As $y = (y_1, \ldots, y_d)$ runs through Z, the sum $\sum_{k=1}^{d} y_k (u+1)^{k-1}$ attains precisely all $|Z| = (u+1)^d$ distinct values $0, 1, \ldots, (u+1)^d - 1$. This implies that, under the total order of the points y in Z by increasing the value of $\sum_{k=1}^{d} y_k (u+1)^{k-1}$, the vector of coefficients of the g_y in the equation corresponding to t is precisely the point $(t^0, t^1, \ldots, t^{(u+1)^d - 1})$ on the moment curve in $\mathbb{R}^Z \simeq \mathbb{R}^{(u+1)^d}$. Therefore, the equations are linearly independent and hence the system can be solved for the $g_y = g_y(a)$ and the desired support $\hat{Y} = \{y \in Y : g_y(a) \neq 0\}$ of $\det(A)$ can indeed be computed in polynomial time. $\square$

We are now in a position to prove Theorem 1.3. By a *randomized algorithm* that solves the nonlinear bipartite matching problem, we mean an algorithm that has access to a random bit generator and on any input to the problem outputs a perfect matching which is optimal with a probability of at least a half. The running time of the algorithm includes a count of the number of random bits used. Note that by repeatedly applying such an algorithm several times and picking the best perfect matching, the probability of failure can be decreased at will; in particular, repeating it n times decreases the failure probability to as negligible a fraction as $\frac{1}{2^n}$ while increasing the running time by a linear factor only.

Theorem 1.3. *For any fixed d, there is a randomized algorithm that, given any positive integer n, any integer weights $w^1, \ldots, w^d$, and any function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ presented by the comparison oracle, solves the nonlinear bipartite matching problem in an oracle-time which is polynomial in n and* $\max |w^k_{i,j}|$.

Proof. As explained in the beginning of this section, we may and will assume that the w^k are nonnegative. First we claim that, with probability at least $1 - \frac{1}{2n}$, we can compute the optimal *objective function value* of the nonlinear bipartite matching problem. To see this, note that the optimal value equals $\max\{f(y) : y \in Y\}$ where $Y = \{w \cdot x : x \in \text{vert}(\Pi^n)\}$ as before, and let $y^* \in Y$ be a point attaining $f(y^*) = \max\{f(y) : y \in Y\}$. Now, using polynomially many random bits, draw independently and uniformly distributed integers from $\{1, 2, \ldots, 2n^2\}$ and substitute them for the $a_{i,j}$. Next compute $\hat{Y} = \{y \in Y : g_y(a) \neq 0\}$ using the algorithm underlying Lemma 5.2 and determine $\max\{f(y) : y \in \hat{Y}\}$. By Proposition 5.1, with probability at least $1 - \frac{1}{2n}$ we have $y^* \in \hat{Y}$ in which event $\max\{f(y) : y \in \hat{Y}\} = \max\{f(y) : y \in Y\}$ is indeed the optimal objective function value.

Next, suppose that $M \subset N = \{(i, j) : 1 \leq i, j \leq n\}$ is any (not necessarily perfect) matching of $K_{n,n}$. Then we can also compute, with probability at least $1 - \frac{1}{2n}$, the maximum objective function value among perfect matchings $M \cup L$ containing M. To see this, let $m := n - |M|$ and consider the subgraph G of $K_{n,n}$ induced by the vertices not matched under M. Then G is isomorphic to $K_{m,m}$ and we have a naturally induced nonlinear bipartite matching problem on G, where the new weight functions $\bar{w}^k$ are simply the restrictions of the w^k to the edges of G, and the new functional $\bar{f}$ on $\mathbb{R}^d$ is defined by $\bar{f}(y_1, \ldots, y_d) := f(y_1 + w^1(M), \ldots, y_d + w^d(M))$. Then the objective function value $f(w^1(M \cup L), \ldots, w^d(M \cup L))$ of any perfect matching $M \cup L$ of $K_{n,n}$ in the original problem equals the objective function value $\bar{f}(\bar{w}^1(L), \ldots, \bar{w}^d(L))$ of the perfect matching L of G in the induced problem. Since $\max \bar{w}^k_{i,j} \leq \max w^k_{i,j}$ and $m \leq n$ we can compute, with probability at least $1 - \frac{1}{2n}$, in polynomial time in n and $\max |w^k_{i,j}|$, the optimal objective function value of a perfect matching of G by the algorithm of the paragraph above applied to G, where the randomized substitutions are taken from $\{1, 2, \ldots, 2mn\}$ (and not from $\{1, 2, \ldots, 2m^2\}$, which would give smaller probability of success). This value is the maximum objective function value among perfect matchings $M \cup L$ containing M.

We claim that the following procedure constructs a perfect matching M of $K_{n,n}$ which is optimal with probability at least $\frac{1}{2}$. Start with $M := \emptyset$ and $i := 1$. While $i \leq n$ iterate: for each edge (i, j) such that j is not matched under M, use the algorithm of the previous paragraph to obtain the maximum objective function value of a perfect matching of $K_{n,n}$ containing $M \cup \{(i, j)\}$; let j_i be the smallest j for which this value is maximal; update $M := M \cup \{(i, j_i)\}$; increment i; and repeat. Output is M.

To prove the claim, let $M^* = \{(1, r_1), (2, r_2), \ldots, (n, r_n)\}$ be the *lexicographically first* optimal perfect matching of $K_{n,n}$, that is, the one such that for any other optimal $M' = \{(1, s_1), (2, s_2), \ldots, (n, s_n)\}$ there is an index $1 \leq h < n$ such that $r_i = s_i$ for all $i < h$ and $r_h < s_h$. For $i = 1, \ldots, n$ let E_i be the random event so that after the completion of the iteration i of the above procedure we have $M = \{(1, r_1), \ldots, (i, r_i)\}$. We prove by induction on i that $\Pr(E_i) \geq (1 - \frac{1}{2n})^i$. For the basis, note that E_1 is the event that the randomized algorithm used during the first iteration that computes correctly the maximum objective function value of a perfect matching containing $\{(1, r_1)\}$, having a probability of at least $1 - \frac{1}{2n}$. For the inductive step, note that $\Pr(E_{i+1}|E_i)$ is the probability that, given that after iteration i we have $M = \{(1, r_1), \ldots, (i, r_i)\}$, the randomized algorithm used during iteration $i + 1$ computes correctly the maximum objective function value of a perfect matching containing $\{(1, r_1), \ldots, (i, r_i), (i + 1, r_{i+1})\}$, which is again at least $1 - \frac{1}{2n}$; as $E_{i+1} \subseteq E_i$; the induction follows by

$$\Pr(E_{i+1}) = \Pr(E_{i+1} \cap E_i) = \Pr(E_{i+1}|E_i)\Pr(E_i) \geq \left(1 - \frac{1}{2n}\right)\left(1 - \frac{1}{2n}\right)^i = \left(1 - \frac{1}{2n}\right)^{i+1}.$$

Now, the probability that the perfect matching M output by the procedure above is optimal is no smaller than the probability that M equals the lexicographically first optimal perfect matching M^*, which is precisely $\Pr(E_n)$ and hence at least $(1 - \frac{1}{2n})^n \geq \frac{1}{2}$ as desired. This completes the proof. $\square$

Acknowledgements

We would like to thank Ilan Newman for his interest and helpful remarks, and the referees for their useful suggestions that improved the presentation. The first author was supported in part by an Irwin and Joan Jacobs Scholarship and by a scholarship from the Graduate School of the Technion. The second author was supported in part by a grant from ISF — the Israel Science Foundation, by the Technion President's Fund, and by the Jewish Communities of Germany Research Fund.

References

[1] R. Chandrasekaran, S.N. Kabadi, K.G. Murty, Some NP-complete problems in linear programming, Oper. Res. Lett. 1 (1982) 101–104.
[2] J. De Loera, S. Onn, The complexity of three-way statistical tables, SIAM J. Comput. 33 (2004) 819–836.
[3] F.K. Hwang, S. Onn, U.G. Rothblum, A polynomial time algorithm for shaped partition problems, SIAM J. Optim. 10 (1999) 70–81.
[4] A.V. Karzanov, On the maximal matchings of a given weight in complete and complete bipartite graphs, Kibernetika 1 (1987) 7–11.
[5] K. Mulmuley, U.V. Vazirani, V.V. Vazirani, Matching is as easy as matrix inversion, Combinatorica 7 (1987) 105–113.
[6] S. Onn, Convex matroid optimization, SIAM J. Discrete Math. 17 (2003) 249–253.
[7] S. Onn, U.G. Rothblum, Convex combinatorial optimization, Discrete Comput. Geom. 32 (2004) 549–566.
[8] S. Onn, L.J. Schulman, The vector partition problem for convex objective functions, Math. Oper. Res. 26 (2001) 583–590.
[9] C.H. Papadimitriou, Polytopes and complexity, in: W.R. Pulleyblank (Ed.), Progress in Combinatorial Optimization, Academic Press, 1984, pp. 295–305.
[10] C.H. Papadimitriou, M. Yanakakis, The complexity of restricted spanning tree problems, J. ACM 29 (1982) 285–309.
[11] J.T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, J. ACM 27 (1980) 701–717.
[12] T. Yi, K.G. Murty, C. Spera, Matchings in colored bipartite networks, Discrete Appl. Math. 121 (2002) 261–277.